Vol. III, No. 4
March/April 1984

satn®

Software Arts Technical Notes from the creators of VisiCalc®

CONTENTS	FEATURING	PAGE
Refinements to Net Present Value and Internal Rate of Return Net present value and internal rate of return are useful tools for comparing alternative investments, but both make certain assumptions. When those assumptions aren't valid—for example, when returns from an investment cannot be reinvested at the cost of capital—NPV and IRR can be misleading. In this article, we show you how to refine your investment analysis by calculating an "adjusted net present value" (and if you have the VisiCalc Advanced Version program, "adjusted internal rate of return").	@NPV @IRR @RATE	2
Sorting Your Worksheet Data We include a program written in the BASIC programming language that can take a VisiCalc print file and sort its data in ascending or descending order. The sorted data is saved in a file that can be included later in reports.	/PF	4
Repeating Keystroke Sequences with Ease How many times have you completed a complicated procedure and then realized that you need to repeat the same series of steps? To save you time and effort when you are entering keystroke sequences that you want to execute more than once, you can use the Keystroke Memory feature of the VisiCalc Advanced Version program. We use a checkbook register to demonstrate this feature.	/K	7
Pointer: Calculating with the # and ! Keys Use these keys to turn cell references and expressions into values when you are performing calculations or after you have entered an expression in a cell.	# !	9
Book Review Julie Bingham reviews *Dynamics of VisiCalc,* by Barry Bayer and Joseph Sobel.		10
Related Products Checklist We list some products that enhance the VisiCalc program.		11

REFINEMENTS TO NET PRESENT VALUE AND INTERNAL RATE OF RETURN

Net present value (NPV) and internal rate of return (IRR) are often calculated to determine if a project will return an acceptable profit or which of two potential projects would earn a higher return. Because of some assumptions built into the calculations, however, simple IRR or NPV determination does not always give a satisfactory answer. We will look at some of the problems with IRR and NPV and then see how the VisiCalc® program can be used to calculate adjusted net present value (ANPV). In addition, we will use the Advanced Version of the VisiCalc program to calculate adjusted internal rate of return (AIRR).

In calculating either IRR or NPV, an assumption is made that the cash flows from the project under consideration can be reinvested at a certain rate. For IRR calculations, that rate is the internal rate of return, while the reinvestment rate used for NPV calculations is usually the firm's cost of capital.

Unfortunately, it is not always realistic to assume that the cash flows can be reinvested at the unadjusted internal rate of return or the cost of capital. When you know that the returns from a project will be reinvested at a rate different from that assumed in the IRR or NPV calculation and when you can make an assumption about what the rate will be, you can use ANPV or AIRR rather than NPV or IRR.

An additional problem with unadjusted NPV or IRR calculations occurs when two projects of different scales or unequal lives are compared. Because of the reinvestment rate assumptions, the NPV and IRR methods may give opposite answers for which project should be undertaken.

Suppose that you are considering two different projects, Project A and Project B. Each project requires an initial outlay of $20,000. The net cash flow realized from Project A will be $10,000 a year for three years, while Project B will return $5000 for years 1 and 2 and $22,500 in year 3. Your firm's cost of capital is 10%.

Figure 1 shows the internal rates of return and the net present values for the projects at a 10% cost of capital. When setting up your sheet, skip the instructions labeled **AVO** (Advanced Version Only) if you are not using the Advanced Version of the VisiCalc program. Those instructions deal with IRR, which is easily computed in the Advanced Version with the @IRR function but which is mentioned here mainly for purposes of explanation and illustration. (SATN®, Vol. I, No. 4, March/April 1982, discusses how to calculate IRR in VisiCalc.) The main focus of this article is on ANPV, and both ANPV and NPV can be calculated easily in any version of the VisiCalc program.

After you have entered the labels shown in Figure 1, set up the upper part of the model—the net cash flows for years 1 through 3 for both projects—by entering the values shown in rows 4, 5, and 6.

How to Use SATN

The VisiCalc program is available on many different personal computers. These machines may have slightly different keyboards. To reduce confusion, SATN will use the symbol ® and the word RETURN synonymously to represent the keys known as ENTER, ENDLINE, and RETURN. The symbol ^ is used to represent the exponentiation operator.

To make VisiCalc documentation more readable, the program was designed so that the colon (:) has the same effect as RETURN in some commands, as when ending the > command and to separate the various elements of the replicate command.

In examples of what to type, type exactly what you see and press RETURN at the end of every line. On some computers, you may find it necessary to add parentheses to some of the expressions included in the text. For example, 3 + 4*5 may become (3 + 4)*5.

Throughout SATN, one typeface represents both VisiCalc input and output (what you should type and what you will see on the screen). The typeface looks like this:

```
This line is typeset in OCR-B.
So is this: +A1+B1*3.25
```

Characters you should type in are printed in rust.

To allow you to type the formulas directly into VisiCalc, the examples use specific coordinates, like A1, rather than the names of the value they represent. When actually typing a formula, you can use the arrows to point rather than using the > command.

SATN is published bimonthly by Software Arts Products Corp., 27 Mica Lane, Wellesley, MA 02181. Address all editorial correspondence to the Editorial Department, SATN, P.O. Box 99, Newton, MA 02162. Requests for special permission or bulk orders should be addressed to the publisher.

Domestic subscriptions: 6 issues $30.00, 12 issues $50.00; outside U.S.A., 6 issues $50.00, 12 issues $90.00. Single-copy price, $6.00. Back-issue price, $7.00. Send subscriptions, change of address, and fulfillment questions to SATN Subscriptions, P.O. Box 97, Newton, MA 02162.

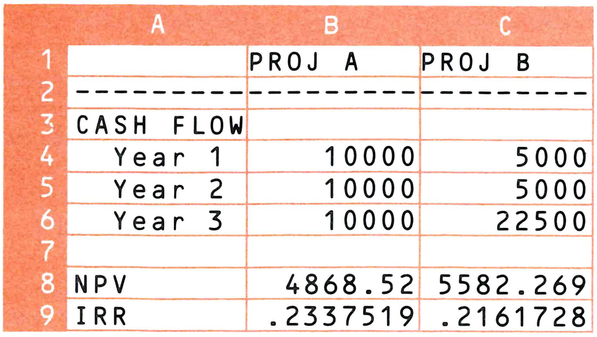
Copyright © 1984 Software Arts Products Corp. All rights reserved. This periodical may not be copied or reproduced in any fashion either in whole or in part without the prior written consent of Software Arts Products Corp.

Software Arts Products Corp. makes no warranty, express or implied, with respect to the quality, accuracy, or freedom from error of the information contained in SATN, the journal for VisiCalc users, and specifically disclaims all liability for any damages resulting from the use of such information.

Unless specifically stated otherwise in any article in this issue of SATN, Software Arts Products Corp. has not examined any products and/or publications identified in SATN to determine their accuracy and/or ability to perform as advertised. Mention of a product and/or publication in SATN does not constitute an endorsement by Software Arts Products Corp. of any such product or publication.

Software Arts is a trademark of Software Arts, Inc. and Software Arts Products Corp. Unless specifically stated otherwise, all references in this issue of SATN to Software Arts shall refer to both companies. DIF is a trademark of Software Arts Products Corp. Keystroke Memory is a trademark of Software Arts Products Corp. SATN is a registered trademark of Software Arts Products Corp. VisiCalc is a registered trademark of VisiCorp.

Apple is a registered trademark of Apple Computer, Inc. IBM is a registered trademark of International Business Machines, Inc. Atari is a trademark of Atari, Inc. Commodore is a trademark of Commodore International. HP-83 and HP-85 are trademarks of Hewlett-Packard Company. TRS-80 is a trademark of Radio Shack division of Tandy Corp.



	A	B	C
1		PROJ A	PROJ B
2	---------	---------	---------
3	CASH FLOW		
4	Year 1	10000	5000
5	Year 2	10000	5000
6	Year 3	10000	22500
7			
8	NPV	4868.52	5582.269
9	IRR	.2337519	.2161728

FIGURE 1. *Unadjusted net present value, which can be calculated in any version of VisiCalc, and internal rate of return, easily calculated in Advanced Version, for Projects A and B at 10% cost of capital.*

Next, find the NPV of the cash flows for the three years at a 10% cost of capital minus the initial $20,000 investment by entering:

```
>B8:@NPV(.1,B4...B6)-20000®/R:C8:RR
```

(If you were setting up this model for repeated use, you would probably want to put the values for cost of capital and initial investment in separate cells. You would then reference these cells in the @NPV expression, rather than including the constants in the expression, as we have done here.)

In Advanced Version VisiCalc, the IRR for each project is found by entering:

```
AVO  >B9:@IRR(-20000,B4...B6)®/R:C9:RR
```

The IRR for Project A is 23.4%, while the IRR for Project B is 21.6%. At a 10% cost of capital, however, the NPV of Project A is only $4868 while the NPV for Project B is $5582. If we use the recommendations given by the NPV method, we would choose Project B, but if we follow the decision rule associated with the IRR method—choosing the project with the higher IRR—we would pick Project A.

Let's see what happens when we use adjusted net present value and adjusted internal rate of return.

The key to determining ANPV and AIRR is calculating a terminal value of the project or projects under consideration. The terminal value is the amount of money that we will have if we are able to invest the cash we net from our project at the new reinvestment rate. It is determined by compounding the cash flows realized each year of the project's life at a new reinvestment rate.

The formulas for computing ANPV and AIRR are

$$ANPV = (TV/(1+k)^{\wedge}n) - Cost$$

and

$$(TV/(1+AIRR)^{\wedge}n) - Cost = 0$$

where
TV = terminal value
k = cost of capital
n = number of periods
$Cost$ = cost of project

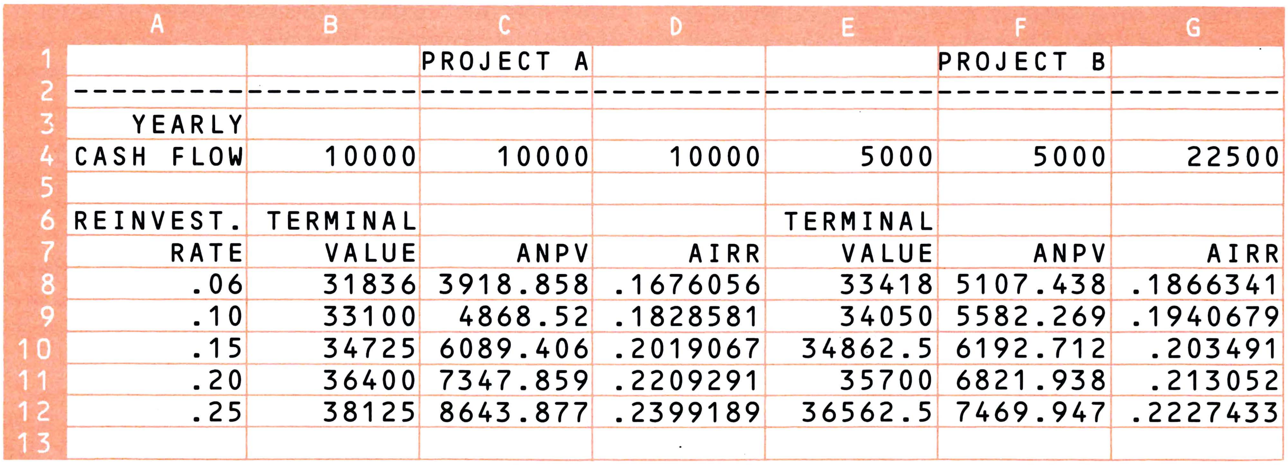
To set up your sheet to look like Figure 2, clear the sheet (/CY) and enter the labels as shown. Then enter the yearly cash flows across row 4 and the reinvestment rates down column A.

Enter the expression to calculate the terminal values as follows:

```
>B8:1+A8*(1+A8*B4+C4)+
                    D4®/R:B9...B12:RRNNN
```

Entries for the ANPV column are:

```
>C8:+B8/(1+.1^3)-20000®/R:C9...C12:R
```



	A	B	C	D	E	F	G
1			PROJECT A			PROJECT B	
2	---------	---------	---------	---------	---------	---------	---------
3	YEARLY						
4	CASH FLOW	10000	10000	10000	5000	5000	22500
5							
6	REINVEST.	TERMINAL			TERMINAL		
7	RATE	VALUE	ANPV	AIRR	VALUE	ANPV	AIRR
8	.06	31836	3918.858	.1676056	33418	5107.438	.1866341
9	.10	33100	4868.52	.1828581	34050	5582.269	.1940679
10	.15	34725	6089.406	.2019067	34862.5	6192.712	.203491
11	.20	36400	7347.859	.2209291	35700	6821.938	.213052
12	.25	38125	8643.877	.2399189	36562.5	7469.947	.2227433
13							

FIGURE 2. *Adjusted net present value and adjusted internal rate of return for Projects A and B at different reinvestment rates. The values in columns D and G—AIRR of Project A and Project B—are calculated in VisiCalc Advanced Version.*

With VisiCalc Advanced Version, you can compute AIRR by entering the following formula:

```
    AVO >D8:@RATE(3,@NA,-20000,
                         B8)®/R:D9...D12:R
```

(Again, when developing a working model, you might want to extract the constants from the preceding expressions.)

Finding the terminal values, ANPVs, and AIRRs for Project B is a simple matter of replicating the expressions used for Project A. If you are not using the Advanced Version of the VisiCalc program, enter:

```
    >B8:/R...C8:E8:NNRRRR
    >E8:/R...F8:E9...E12:RRNNNR
```

If you are using the Advanced Version, enter:

```
   AVO >B8:/R...D8:E8:NNRRRRR
   AVO >E8:/R...G8:E9...E12:RRNNNRR
```

Remember that at a 10% cost of capital, the unadjusted NPV for Project B was higher than that for Project A, while the unadjusted IRR was higher for Project A. Notice that this is not the case with ANPV and AIRR. As Figure 2 shows, ANPV and AIRR rank the projects in the same order at all reinvestment rates. The crossover point—the point at which both projects look equally attractive from the standpoint of return on our investment—is just over 15%. If the cash flows from the project can be reinvested at 6%, 10%, or 15%, Project B should be chosen, but Project A will give a better return if the reinvestment rate is 20% or 25%.

We have seen that calculating ANPV or AIRR gives more complete information than NPV or IRR if certain assumptions inherent in NPV and IRR calculations are not met. Specifically, when the cash flows will not be reinvested at the cost of capital or when two projects of different scales or lives are compared, ANPV or AIRR will generate more accurate information than NPV or IRR. The discrepancies that can occur when comparing two projects with NPV and IRR do not occur when ANPV and AIRR are used.

Since the two methods give the same answer for which of two projects is preferable, it is not necessary to compute both of them. While we have looked at both ANPV and AIRR, AIRR has been mentioned mainly for illustrative purposes. ANPV can be computed easily in all versions of VisiCalc, and it is just as useful for decision making as AIRR.

SORTING YOUR WORKSHEET DATA

One way to extend the power of the VisiCalc® program is to use utility programs that operate on VisiCalc files. In the September/October 1983 issue of SATN® we reviewed Edouard Desautels' *Advanced Uses of VisiCalc for the IBM Personal Computer*, which includes many such utilities written in the BASIC programming language. In this article we include one of them, modified slightly. The program takes a VisiCalc print file (one saved with the Print command), sorts it alphabetically or numerically on a designated column, and saves the sorted image in a file. That file can be printed later or included in a document.

Desautels' book (published by Wm. C. Brown, Dubuque, Iowa) includes two programs, SORT1 and SORT2, that take a VisiCalc print file and sort it. SORT2 allows you to specify the column you want to use as the sorting key and allows you to sort in either ascending or descending order. It displays the sorted image on the screen. We have modified the program, with the permission of the publisher, creating SORT3. Our version allows you to save the sorted output in a file. Figure 3 is a listing of SORT3. Subscribers to SATN are authorized to make copies of the SORT3 program for their personal use or for use in their businesses. Reproduction of SORT3 for sale or for distribution to nonsubscribers is prohibited.

(To make it easier for those who want to compare this program with SORT2, we have not changed any line numbers; all lines we added have a line number that ends in "5." The only other changes we made are in lines 440 and 500, where we added a second "PRINT" statement, which sends the sorted output to a file.)

Briefly, here's how SORT3 works:

You specify a print file to be sorted (line 80), the column width and key (lines 90, 100), and the name of the sorted file (105). SORT3 uses the column width in deciding "how far" into a row to go to get to the key column and how many characters to read when it gets there (the program, therefore, will not work with the Advanced Version of the VisiCalc program when you have specified different column widths).

SORT3 opens the print file and reads it (120–190). It then sorts the lines by comparing them one at a time (200–380).

After you specify whether you want the file to be sorted in ascending or descending order (400), the data is displayed on the screen and saved in a file (440 or 500).

A Demonstration

We have sorted the print file shown in Figure 4, which records sales and profits for a firm from 1975 through 1983. In the first sort, Figure 5, we used "Sales," the second column, as the key and sorted in descending order. The second sort, Figure 6, shows the data organized by descending profits, that is, sorted using the third column as the key.

```
10 'Program SORT3
20 'SORT3 extracts the specified column as the sort "key",
30 '  sorts the print file according to the key,
35 '  and stores the sorted image in a new file
40 'Author: ejd, Version 1,
        Copyright (C) 1983 Wm. C. Brown Company Publishers.
45 'Modified by RH, Software Arts, on 12/15/1983
50 DEFINT A-Z
60 PRINT "Program SORT3 is active."
70 DIM L$(254), I(254), Y$(254)
80 INPUT "Type full name of source file: ", Z$
90 INPUT "What is the column width? ",COL
100 INPUT "Which column controls the sort (type a 1 for A,
                               2 for B, etc.)? ", C
105 INPUT "Type full name of sorted file: ", S$
110 FIRST = 1 + COL * (C-1)  ' assume all columns are
                                        COL characters wide
120 OPEN Z$ FOR INPUT AS #1
125 OPEN S$ FOR OUTPUT AS #2
130 FOR I = 1 TO 254
140    IF EOF(1) THEN 200
150    LINE INPUT #1, X$ ' READ LINE OF PRINT FILE
160    L$(I) = MID$( X$, FIRST, COL )
170    Y$(I) = X$
180    I(I) = I
190 NEXT I
200 MM = I-1  ' number of rows
210 M = MM
220 M = INT(M/2)
230 IF M = 0 THEN 390
240 K = MM - M
250 J = 1
260 I = J
270 L = I + M
280 IF L$(I) <= L$(L) THEN 370
290 T$ = L$(I)
300 L$(I) = L$(L)
310 L$(L) = T$
320 T = I(I)
330 I(I) = I(L)
340 I(L) = T
350 I = I - M
360 IF I >= 1 THEN 270
370 J = J + 1
380 IF J <= K THEN 260 ELSE 220
390 ' sort done; print result
400 INPUT "Type 1 for ascending order, or 2 for
                               descending order: ",ORDER
410 IF ORDER = 2 THEN 480
420 FOR I = 1 TO MM
430     J = I(I)
440     PRINT Y$(J)     :      PRINT #2, Y$(J)
450 NEXT I
460 GOTO 520
470 'descending order
480 FOR K = MM TO 1 STEP -1 'step down by 1s
490    J = I(K)
500    PRINT Y$(J)      :      PRINT #2, Y$(J)
510 NEXT K
520 CLOSE
530 END
```

FIGURE 3. *SORT3, a modified version of SORT2, which appeared in Edouard Desautels'* Advanced Uses of VisiCalc for the IBM Personal Computer. *The program sorts a VisiCalc print file and saves the sorted image in a file for later use. This program is written for the IBM Personal Computer and may have to be modified to run on other machines. Check your owner's manual for information on how to run BASIC programs on your system.*

```
 Year       Sales      Profit
------------------------------
  1975       234        112
  1976       254        130
  1977       203        115
  1978       270        127
  1979       265        145
  1980       280        190
  1981       295        165
  1982       320        180
  1983       302        155
```

FIGURE 4. *The original VisiCalc print file, which details sales and profits in the years 1975 through 1983. The source worksheet had been formatted with /GFL.*

Limitations of the Program

This BASIC program sorts the characters according to their representation in ASCII, the standard microcomputer alphanumeric coding system. In ASCII, the number "0," for example, is character number 48, "1" is 49, "2" is 50, and so on. Letters start at character number 65, which corresponds to capital "A"; "B" is 66, and so on. Lowercase letters start at number 97, for "a." So, in ASCII, all numbers are "less in value" than letters, and all uppercase letters are less in value than lowercase letters.

```
 Year       Sales      Profit
------------------------------
  1982       320        180
  1983       302        155
  1981       295        165
  1980       280        190
  1978       270        127
  1979       265        145
  1976       254        130
  1975       234        112
  1977       203        115
```

FIGURE 5. *The data shown in Figure 4 was processed by the BASIC sorting program to organize the data by sales.*

Since letters have a greater ASCII value than numbers, labels will end up below numbers if the sort is made in ascending order. In the examples shown, the labels *Year*, *Sales*, and *Profit* would have appeared below all the values if we had sorted in ascending order. It would, however, be a simple matter to modify the BASIC program so that sorting starts on the second (or third or fourth...) row, leaving any labels in place. You could even have the program "ask" you when you run it how many rows to leave intact.

```
 Year       Sales      Profit
------------------------------
  1980       280        190
  1982       320        180
  1981       295        165
  1983       302        155
  1979       265        145
  1976       254        130
  1978       270        127
  1977       203        115
  1975       234        112
```

FIGURE 6. *Here the data is sorted by profit.*

Spaces are treated as characters. So, for example, if labels are right-aligned, they will be seen by the program as beginning with a number of spaces. Since the space "character" has a lower ASCII value than letters, right-aligned labels that are short will have a lower value than longer ones, regardless of the first letter.

Advanced Uses of VisiCalc for the IBM Personal Computer also includes BASIC programs that sort DIF™ files, thereby allowing you to take VisiCalc data, save it in a DIF file, sort it, and load the sorted file back into the sheet.

There are a number of utilities on the market similar to the BASIC program described here. Check them out; they may be just what you are looking for to extend the power of the VisiCalc program.

BACK ISSUES AVAILABLE

Back issues of SATN are now available. The cost is $7.00 per copy for orders mailed within the USA. If your order is to be mailed outside the USA, add $2.50 for shipping and handling to the full price of the order.

To order back issues, please send your name and address, a list of the specific issues of SATN you want (by Volume and Issue numbers), and a check or money order payable to SATN to:

SATN Back Issues
P.O. Box 99
Newton Lower Falls, MA 02162

REPEATING KEYSTROKE SEQUENCES WITH EASE

The Keystroke Memory™ feature of the VisiCalc® Advanced Version program, currently available for the Apple IIe and Apple /// personal computers, lets you create a one-character abbreviation for a sequence of keystrokes. These sequences are called "macros." Unlike those in some other spreadsheets, these macros are not merely a series of labels entered in a defined area on the worksheet. Rather, the VisiCalc Advanced Version program's macros are internal commands that can be easily stored while you are executing them or stored and executed later.

The Keystroke Memory feature facilitates repeating procedures, whether they involve simple or complex sequences. You can use this feature to format and label frequently used worksheets, compile printing commands, save and load files, perform complex calculations, or convert a series of expressions into values. In fact, you can use this feature to chain any number of commands that you want to execute together.

For instance, most of you probably have a checking account and have experienced the frustration of trying to keep your checkbook register in order. From incorrectly numbering the checks to miscalculating the balance, you are frequently vulnerable to making errors. To minimize the chances for such errors and to make your checkbook maintenance easier, you can use the Keystroke Memory feature. We will use a checkbook register model to demonstrate how this feature can automate functions such as recording paycheck deposits and determining balances after checks.

If you have the VisiCalc Advanced Version program, you can build the model with us. Even if you don't have the Advanced Version, you may want to read this article to pick up ideas on how to set up your checkbook register with the VisiCalc program.

A Sample Use of Macros

Begin the automated checkbook register by setting up the labels and dashed line in rows 1 through 3 on your worksheet, as shown in Figure 7.

Before creating any macros, we'll describe the two commands you can use to define keystroke sequences and the differences between these commands.

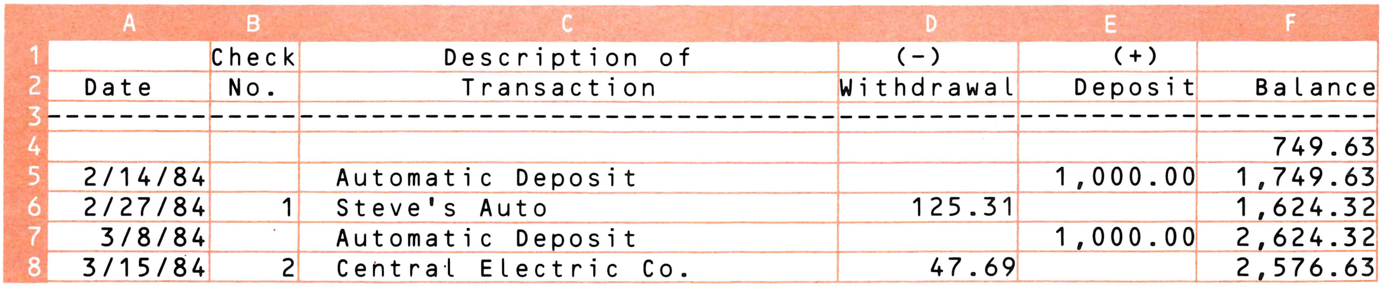
We will use the /K= command to define keystroke sequences. You could also use the Control-K= command, which, unlike the /K= command, performs a sequence as you define it. (On the Apple IIe, this command is Control-S=.) Because the /K= command does not execute a sequence as you are entering the keystrokes, you can edit the sequence while you are creating it by backspacing to delete characters or by using the Control-E command. After you have created a macro, you can always edit it by typing /KE and the macro name and then making the necessary changes.

The /K= and Control-K= commands allow you to keep coordinates relative within your expressions by using special characters for certain keys (see Table 1).

When you are defining keystroke sequences, you can use either uppercase or lowercase letters for commands. The VisiCalc program does not distinguish between the two when you are using the Keystroke Memory feature.

We are now ready to build two macros that can be used to perform some typical functions in a checkbook register. Your individual needs for your register may differ from ours. Nevertheless, follow through as we develop macros for automatic paycheck deposits and balances after checks. You can use the information we provide to help you build macros that are tailored to your needs.

We will begin our first macro at the Date column. Though the Keystroke Memory feature is a handy tool, it cannot predict the date on which you will make transactions. You must supply the date for each entry in your register.



	A	B	C	D	E	F
1		Check	Description of	(-)	(+)	
2	Date	No.	Transaction	Withdrawal	Deposit	Balance
3	---------	-----	------------------------------	----------	----------	----------
4						749.63
5	2/14/84		Automatic Deposit		1,000.00	1,749.63
6	2/27/84	1	Steve's Auto	125.31		1,624.32
7	3/8/84		Automatic Deposit		1,000.00	2,624.32
8	3/15/84	2	Central Electric Co.	47.69		2,576.63

FIGURE 7. *When you set up your checkbook register, change your column widths so that your worksheet appears as above. Using the /GCC command, make column B five spaces, column C thirty spaces, and columns D, E, and F each ten spaces. Also, center the column headings with the /GALC command. To make the entries in this register, we used macro A in rows 5 and 7 and macro C in rows 6 and 8. The transaction dates were entered manually.*

Character	Function
^U	Up arrow
^D	Down arrow
^<	Left arrow
^>	Right arrow
^R	Return
^P	Pause and display a message
^K	Link one sequence to another

TABLE 1. *Use these special characters (a caret followed by a letter or symbol) when you are defining keystroke sequences with the /K= command or when editing an existing sequence. You can link sequences by typing ^K followed by the one-letter abbreviation of another macro.*

Since many people have their paychecks automatically deposited into their checking accounts each pay period, a useful macro is one that records the deposit and calculates the new balance. For this model, we will use $1000 as the amount of deposit.

The automatic paycheck deposit macro begins by moving the cursor to the Description of Transaction column (accomplished by ^>^>). It will enter the label "Automatic Deposit" and then place the cursor in the Deposit column (^>^>), where the deposit of $1000 will be recorded. The macro will then prepare to calculate a new balance by placing the cursor in the Balance column (^>).

Calculating the balance is a relatively easy procedure now that the cursor is in the proper column. The deposit macro will add the previous balance (+^U) to the deposit (+^<) and enter the result (^R). As a final step, this macro will return the cursor to the Date column in the next row so that you are ready to begin a new entry (^D^<^<^<^<).

Now you can create the automatic paycheck deposit macro. Label it "A" for automatic deposit and enter:

```
/K=A^>^>Automatic Deposit^>^>
        1000^>+^U+^<^R^D^<^<^<^<
```

Remember that when you use this macro, you must begin with the cursor in the Date column.

Try using macro A now. Enter 2/14/84 in the Date column and then type **Control-K A**. (Be sure to press the CONTROL and K keys simultaneously.) If you are using an Apple IIe, type Control-S A.

The second keystroke sequence that we will define determines the balance after each check has been written. Since the date, payee, and amount will most likely differ for each entry, you must enter this information yourself. But once you have entered an amount in the Withdrawal column, the macro will do the work for you.

The keystroke sequence will start with a pause message asking if the cursor is in the proper column. The Pause option (^P) of the Keystroke Memory feature allows you to display a message on the prompt line during a keystroke sequence and then resume the sequence by pressing RETURN. Or, if you want to stop a sequence after a pause message without executing the rest of the macro, you can press Break. (See your VisiCalc Advanced Version program manual for a description of what keys to use for Break on your machine.) Note that to put a question mark in a pause message, type Control-Q. A colon ends the message.

Next, the macro will move the cursor from the Withdrawal column to the Balance column (^>^>), subtract the withdrawal amount from the previous balance (+^U–^<^<), and enter the result (^R).

The third part of the macro, which numbers the check, begins by moving the cursor from the Balance column to the Check Number column (^<^<^<^<). To ensure that the checks are numbered chronologically, regardless of how many deposits (which don't require numbers) you make, use the @MAX function to add 1 to the largest number in the Check Number column (column B). The expression that accomplishes this is 1+@MAX(B4...^U).

Finally, the macro will move the cursor from the Check Number column to the Date column in the following row (^D^<) so that you are ready for the next transaction.

Combine the four parts of the macro, name it "C" for check, and enter:

```
/K=C^PAre you in the Withdrawal
        columnControl-Q:^>^>+^U-^<^<^R
            ^<^<^<^<1+@MAX(B4...^U)^D^<
```

Now try using macro C. Enter 2/27/84 in the Date column, Steve's Auto in the Description of Transaction column, and 125.31 in the Withdrawal column. Then invoke the macro by typing **Control-K C**.

You can always review the macros you have created by typing /K RETURN and then pressing the right arrow key. Do that now to review macros A and C. The keystroke sequences will appear on the edit line in the order in which they were created.

Modifying Macros

You can readily adapt these two macros to meet your individual needs by editing the keystroke sequences we have developed. For instance, you can edit macro A (automatic deposit) to reflect your actual deposit. Simply type /KEA, move the cursor to the dollar amount, and replace it with the amount of your deposit.

Similarly, you can define additional keystroke sequences based on the ones in this model. For example, you can set up a macro that assigns a check number, enters the name of your electric, gas, or oil company as the payee, and then places the cursor in the Withdrawal column so that you can enter the amount of your check.

You can develop as many as 26 different keystroke sequences (A-Z), each of which can be as long as 123 characters. Therefore, by linking sequences, you can execute an unlimited number of commands with just a few keystrokes.

MACROS WITHIN MACROS

You may have noticed that in the article, keystroke sequence C contained the expression 1+@MAX(B4...^U). This expression is used to generate the chronological check numbers in the automated checkbook register. As with any other repeated sequence of characters, the 1+@MAX(B4...^U) expression could be assigned a one-letter abbreviation and used as a macro within other macros.

For instance, you could name the expression "N" for number and enter:

```
/K=N1+@MAX(B4...^U)
```

Then each time this keystroke sequence is needed in another macro, you would enter ^KN for the sequence.

For example, keystroke sequence C (check balance) would be entered as:

```
/K=C^PAre you in the Withdrawal
    column Control-Q:^>^>+^U-^<^<^R
                    ^<^<^<^<^KN^D^<
```

Whenever you know that you will be repeating a keystroke sequence in several macros, you may want to define the sequence as a separate macro. By doing so, you can combine macros and save yourself work since you will not have to enter the entire keystroke sequence each time.

POINTER: CALCULATING WITH THE # AND ! KEYS

Two frequently overlooked keys that can be helpful when performing calculations in the VisiCalc® program are the pound sign (#) and the exclamation point (!). By using these special characters, you can turn cell references and expressions into values.

If, while typing an expression, you press # after you have referenced a cell, the value of that cell replaces the reference to it on the edit line. For instance, if A4 contains the value 32 and B4 contains the value 56, then typing +A4+B4# would display +A4+56 on the edit line.

Pressing ! while entering an expression causes the calculated value of the entire expression to be displayed on the edit line. Using the above example, if you were to type +A4+B4!, the edit line would display 88.

Note that these conversions into values are not reversible; once the value appears, the underlying references cannot be restored without retyping them.

The # key has a second use. If you type # in a cell that contains an expression, the value of that expression appears on the edit line. If you then type RETURN, the value replaces the expression as the cell entry. (If you want to retain the expression after you have typed #, press Break instead of RETURN.)

Turning expressions and references into values allows you to:

- Copy the current value of one cell into another cell. Place the cursor in the cell into which you want to copy the value and type +. Then indicate (by either moving the cursor or typing the coordinate) the cell from which the value is to be copied and type #RETURN. Even if the original cell's value changes, the copied value will not.
- Keep your model free of work, such as intermediate calculations, that may clutter your worksheet or that may not be necessary or desirable to display.
- Copy long numbers using # without having to type each digit manually, thereby preventing entry errors and saving time.
- Display subtotals on the edit line with ! before completing an entire calculation.
- Speed recalculation of the worksheet by simplifying an expression with # or ! so that it needn't be evaluated each time the cell containing the expression is referenced. This procedure of converting expressions into values can be used to make your data similar to how it would appear in a DIF™ file. But unlike a DIF file, the data converted by # or ! does not have to be saved and loaded and will not lose the formats you have set.

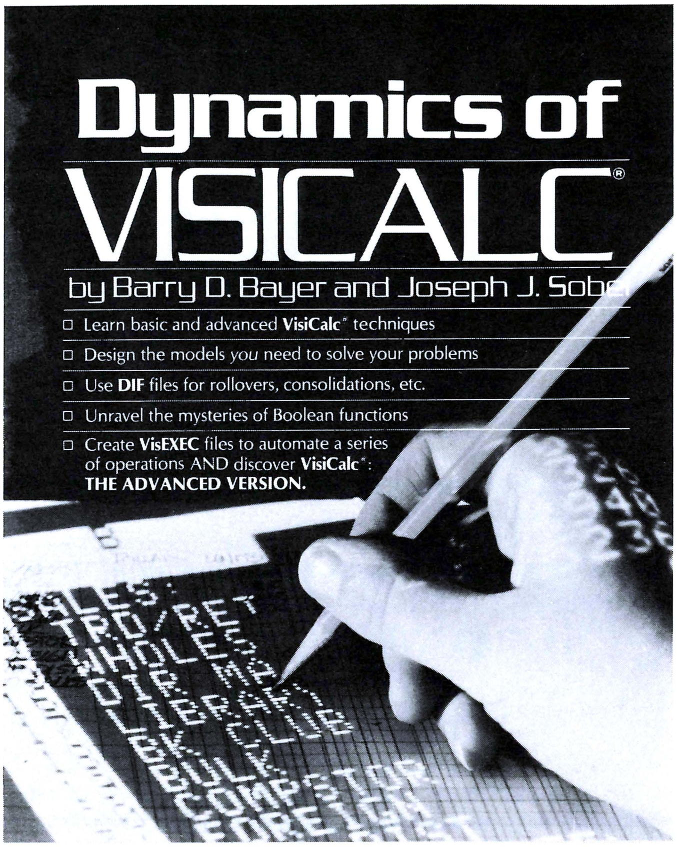
BOOK REVIEW

Reviews published in SATN are the opinions of the reviewers, not of Software Arts, and may not be construed as official positions or endorsements by Software Arts.

Dynamics of VisiCalc

by Barry D. Bayer and Joseph J. Sobel. Dow Jones-Irwin, Homewood, IL, 189 pp., $19.95.

Reviewed by Julie Bingham



Dynamics of VISICALC®

by Barry D. Bayer and Joseph J. Sob

☐ Learn basic and advanced **VisiCalc**® techniques

☐ Design the models *you* need to solve your problems

☐ Use **DIF** files for rollovers, consolidations, etc.

☐ Unravel the mysteries of Boolean functions

☐ Create **VisEXEC** files to automate a series of operations AND discover **VisiCalc**®: **THE ADVANCED VERSION.**

Photograph by Dana Sigall

Dynamics of VisiCalc is something of an "advanced topics" manual with a welcome emphasis on good design practices. Though the book lacks a clear focus, it includes many useful suggestions and tips about VisiCalc® commands and functions, including some little-documented features.

It is not obvious who the intended readers are. The book is too difficult for novices (who are referred to the tutorial in the VisiCalc manual), yet it covers subjects that most seasoned users would already know, such as setting global formats and the @SUM function. The book is most appropriate for the frequent user who can skim through familiar topics and still gain some new insights into solving business problems.

Co-authors Barry Bayer and Joseph Sobel learned the VisiCalc program "out of a need to use a computer to manipulate data from our respective professions" (Bayer is a lawyer, Sobel an audit partner). Their backgrounds give them a business perspective that many computer-book authors lack. This perspective should be welcomed by non-technical readers who need some guidance in how to use VisiCalc to make their everyday problems easier to solve.

Perhaps the most important contribution *Dynamics of VisiCalc* makes to the business user is its strong emphasis on good planning and design techniques. The authors discuss the importance of designing on paper, keeping constants out of formulas, and separating input and output sections. They present several hints on how to design a template so an inexperienced person can use it. This is especially important for the business user who wants to build templates for use year after year.

Bayer and Sobel cover a rather wide range of other topics, from the @IF function to hardware add-ons. One of the most interesting sections in the book documents a feature that can be a real boon to serious VisiCalc users. It involves creating files—called "VisEXEC files" by the authors—that automatically execute a series of commands when they are loaded in combination with an existing worksheet. The authors note that this "feature" is not supported (or recommended) by Software Arts because the results may vary among versions of the program and may not be possible at all in future versions. Bayer and Sobel point out, however, that this feature, if possible on your system, can save a considerable amount of time.

A VisEXEC file is like a small program designed specifically for the application you are using. It consists of a series of commands entered as a column of right-justified labels. Each label begins with the Go To command followed by a cell address and a colon (for example, >A1:) and is followed by a regular VisiCalc command. The labels are printed to disk using the Print File command (/PF). They are then executed by the VisiCalc program when loaded on top of an existing template with the Storage command (/S). For example, if a print file containing the label >A30:/IR is loaded on top of an existing template, the VisiCalc program would go to cell A30 and insert a row.

The example given in this section illustrates how VisEXEC files can be used to perform a "rollover," a technique that updates accounting records. The presentation is especially effective because it follows an example that illustrates how DIF™ files can be used for rollovers. The reader learns two alternatives for solving a common business problem that is not easily adapted to electronic spreadsheets.

(Editor's Note: The VisiCalc Advanced Version program contains a feature called Keystroke Memory™, which is a much simpler and more powerful technique for executing a series of commands. See the article about Keystroke Memory in this issue.)

On the negative side, *Dynamics of VisiCalc* contains some extremely confusing sections that include some rather convoluted examples. Chapter 2 introduces a long @IF formula to compute marginal tax rate:

```
@IF(C10<C6,(E6*((100-C4)/100))*
        ((1+(((100-C4)/100)*(C3/100)))^(C6-C10)),0)
```

The explanation of this formula is followed by the remark, "The design of the rest of this particular template will be discussed in Chapter 4, with the introduction of the "@IF" function." The authors should have either used a simpler example (not involving @IF) in Chapter 2, or moved the introduction of @IF up from Chapter 4.

The book includes a long and confusing description of Boolean logic that would benefit from better examples and a simpler set of terms. However, a major section on the new features and commands of the VisiCalc Advanced Version program is easy to read and understand, and it illustrates the new VisiCalc options quite clearly.

On the whole, *Dynamics of VisiCalc* presents some useful and interesting information for the serious VisiCalc user. The authors explain the importance of good design practices and give some tips on how to design a model so it's easy to use and modify. Though there are some confusing explanations of individual commands or functions, most of the examples are clear and helpful.

Julie Bingham is Marketing Manager at Forum/Nevison Executive Computing in Concord, MA.

RELATED PRODUCTS AND PUBLICATIONS CHECKLIST

The information contained in this column has been garnered from a variety of sources including dealers, press releases, manufacturers, and advertising.

The first criterion for including any article, book, or newsletter in this listing was whether we could find a hard copy in our library. The second was that the item appeared to be useful. We have included only articles that contain information that can work for you, either in the form of models that you can use or reviews of products that complement the VisiCalc® program.

No one at Software Arts has reviewed or examined any of these products to determine whether they perform as claimed or as listed below. An inclusion in this listing does not constitute an endorsement by Software Arts of that product. We also wish to point out that if any of the products listed alter, patch, or copy the VisiCalc program, they could be in violation of your user licensing agreement and/or the copyright.

Since we have not reviewed any of the products, we cannot provide you with any more information than is contained in the list. If you want to learn more or you wish to purchase any of the products listed, contact your dealer first. Although we have identified distributors in some listings, many distributors do not accept orders from individuals for single copies of software or pieces of hardware.

SOFTWARE

Only software that claims to complement the VisiCalc program will be listed.

Templates

Financial Management Models for the Service Firm Package

- Runs on Apple II+ with 48K, Apple IIe, Apple /// with 128K, and IBM PC

A series of 11 VisiCalc templates for such fields as engineering, advertising, architecture, and consulting. The models help to forecast cash needs, budget projects, estimate fees, and determine billing rates. The models are set up to be adapted to the individual firm's needs. A 175-page manual is included.

Distributed by PRO/PAC, Inc., 14925a Memorial Drive, Suite 207, Houston, TX 77079

Suggested retail price: $295

Library Micro Clearinghouse

- Runs on Apple II, Apple II+, and Apple IIe

A non-profit, national clearinghouse established to promote effective library management through the open exchange of library application microcomputer templates. The initial software being used is VisiCalc and DB Master.

Distributed by Eric S. Anderson, Microcomputer Libraries, Freeport, IL 61032

Suggested price: $5 per single-disk template; $7.50 per double-disk template

Taxcalc

- IBM PC, IBM XT, Apple II+, Apple ///+, TRS-80 Models II, III, IV, and 16

Can calculate taxes due, check manual returns, and plan taxes. Input lines follow IRS forms. Computations include capital gain deduction, capital loss limitation, contribution limitations, medical expense limitations, and two-income married couple deduction.

Distributed by Taxcalc, Inc., 4210 W. Vickery Blvd., Fort Worth, TX 76107

Suggested retail price: $125; $50 for yearly updates (California income-tax template is $100)

Graphics

CALCU-PLOT

- Runs on Apple II and Apple IIe with 48K and one or two disk drives (printer is optional)

Solves equations and plots solutions graphically one at a time or in sets. Data is accepted from keyboard, disk files, and print-format VisiCalc files.

Distributed by Human Systems Dynamics, 9010 Reseda Blvd., Suite 222, Northridge, CA 91324

Suggested retail price: $150

Fast Graphics

- Runs on IBM PC

Numeric information from VisiCalc is graphically displayed with line charts or comparative and stacked-bar charts using up to six different sets of data. Titles and footnotes may be added. Customization of charts and special drawing are possible using the graphic editor with draw mode. Charts, graphs, and screens of text can be automatically displayed with the "Slide Show" feature.

Distributed by Innovative Software, 9300 W. 110th St., Suite 380, Overland Park, KS 66210

Suggested retail price: $350

HAVE YOU MOVED?

If you have, let us know. To keep your issues of SATN coming to you regularly, we need to know your current address. Send a card or letter, indicating both your old address and your new one, to:

SATN Changes
P.O. Box 99
Newton Lower Falls, MA 02162

Remember to allow about 4 weeks for the change to take effect.

Communications

Teleminder

- Runs on Apple II, Apple II+, Apple IIe, IBM PC, and IBM XT

Allows automatic, unattended access and retrieval of data from the Dow Jones News/Retrieval Service. The optional Tele-Pak I utility enables retrieved data to be transferred directly into the VisiCalc program.

Distributed by Teleware Inc., P.O. Box 729, Pine Brook, NJ 07058

Suggested retail price: $195 Teleminder; $30 Tele-Pak I

Other Software

TDM

- Runs on Apple II with 64K, 80-column board, and 250K bytes of disk storage; Apple /// with hard disk or three floppy disks

A data management and reporting system for large databases and complex reporting requirements. Can be used with text processing formatters to produce reports or personalized form letters with automatic substitution based on database contents. TDM produces files that may be used by other programs, including VisiCalc. The program supports relational databases.

Distributed by: Pascal Systems, Inc., 830 Menlo Ave., Suite 109, Menlo Park, CA 94026

Suggested retail price: $295

PUBLICATIONS

Books

Executive VisiCalc for the Apple Computer, by Roger E. Clark (Addison-Wesley, 1983). Supplements the VisiCalc user's manual by providing additional instruction on using logic functions to develop complex models and by giving sample business models.

54 VisiCalc Models: Finance, Statistics, Mathematics, by Robert Flast (Osborne/McGraw-Hill, 1983). A book of models for those who use VisiCalc in the financial, statistical, or mathematical fields.

The VisiCalc Applications Book, by Jack Grushcow (Reston Publishing Co., 1983). Demonstrates how to use VisiCalc for financial statement analysis, forecasting, budgeting, cash management, portfolio management, and other business functions using the Apple II, IBM PC, and TRS-80 microcomputers.

Visicalc Programming: No Experience Necessary, by Shaffer and Shaffer Applied Research and Development, Inc. (Little, Brown and Company, 1983). A VisiCalc tutorial consisting of a 236-page manual, an instruction disk, and a command reference card. Versions are available for IBM PC, Apple II, Apple II+, Apple IIe, Atari 800, and TRS-80 Model III.